REMARKS

The present application presents an approach to formalizing the parameterization of data flow graphs to allow runtime parameters. Runtime parameters allow an application builder to defer the value of a parameter setting (*e.g.*, the key parameter of a sort function, file names, record formats, transform functions, *etc.*) to runtime (*i.e.*, the time an application is executed on a computer system). The values of runtime parameters may be supplied by the end user, or be derived from a combination of other runtime parameters or objects stored in an object repository, or be determined from a default setting.

Runtime parameters add flexibility to an application. Additional flexibility is achieved by using those parameters to compute metadata (data formats or types, and program logic or transforms) on demand. Types and transforms may be synthesized from other types and transforms, user-supplied parameter values, and stored objects (*e.g.*, from a repository). This makes it possible to build "generic" applications that work on input data of any type, or that produce data through a series of transforms whose construction is controlled, directly or indirectly, through runtime parameter values.

One embodiment described in the application includes conditional components that permit changes to a graph structure (*and hence the application represented by the graph*) based on parameter values and computed metadata. Each component of a graph has a condition that controls whether or not that component will appear in the graph at runtime. The condition can be computed directly or indirectly through runtime parameters. Conditional components can be used to optimize or specialize graphs.

Claim Rejections – 35 USC § 102

All pending claims 1, 3-5, 8-9, 11-14, 16-18, 21-22, 24-27, 29-31, 34-35, and 37-39 stand rejected under 35 U.S.C. 102(a) as being allegedly anticipated by Stanfill et al. (U.S. Patent No. 5,966,072, hereinafter "Stanfill '072").

Independent claims 1, 14, and 27

Stanfill '072 teaches a method and apparatus by which a graph can be structurally altered so as to be able to invoke computations directly (and describes technology owned by the assignee of the present invention). However, contrary to the position taken in the Office Action, Stanfill '072 fails to teach or suggest each of the elements (a) - (g) of claim 1, 14, or 27.

The Examiner has cited various portions of the Stanfill '072 as disclosing particular elements of claim 1. However, these portions, interpreted in their appropriate context, do not support the Examiner's arguments.

For example, the description in Stanfill '072 at column 13, line 53 to column 14, line 2 (cited by the Examiner as disclosing elements (a) and (b) of claim 1) discloses that a graph can have its nodes and links scanned in order to enable the graph to invoke computations directly, by inserting appropriate file and communication functionality into the graph (thus structurally altering the executable version of the graph). Importantly, Stanfill '072 makes no mention, directly or indirectly, of runtime parameters having values determined at runtime execution of the graph. Instead, as described at column 8, line 56 to column 9, line 4, a driver 200 may create file and process vertices and links when a user 202 provides necessary information. Thus, the input from the user 202 is at *design* time, not runtime – there is no *deferral* of value determination for parameters until *runtime*. Stanfill '072 does not teach or suggest "(a) programmatically retrieving a runtime parameter for the graph at runtime execution of the graph, the runtime parameter having a value defined as determinable at runtime execution of the graph" or "(b) determining whether the value for the runtime parameter is to be provided by user input or is to be externally supplied programmatically."

The Examiner cited column 8, lines 33-55 of Stanfill '072 as disclosing element (c) of claim 1, and column 8 lines, 14-67 as disclosing element (d) of claim 1. Instead, the description in Stanfill '072 at column 8, lines 15-55 discloses that a user 202 can interact with a user interface 204 to build and control the execution of a graph. Again, this is at design time, not runtime. Nothing in Stanfill '072 teaches or suggests either "(c) displaying a prompt to a user for receiving user input for every runtime parameter so determined to be provided by user input" or "(d) retrieving any externally supplied value for every runtime parameter determined to be externally supplied programmatically."

The description in Stanfill '072 at column 17, lines 17-23 (cited by the Examiner as disclosing inherently element (e) of claim 1) discloses that the process is iterative, in the context of showing (starting at FIG. 10) how a graph is processed by the Stanfill '072 approach. Again, there is no discussion of parameters, let alone "(e) determining a final parameter value based on one of the user input to such prompt or such externally supplied value or a default value."

The description in Stanfill '072 at column 16, line 67 to column 17, line 9 (cited by the Examiner as teaching inherently element (f) of claim 1) discloses how state information of components inserted into a graph can be examined and changed (in the example given, two "Copy" programs are marked as "runnable", and communication methods are chosen for those links attached to runnable processes). There is no teaching or suggestion about "(f) modifying the application represented by the graph *using the final parameter value* as the value for the runtime parameter."

The description in Stanfill '072 at column 17, line 30-43 (cited by the Examiner as disclosing inherently the step (g) of claim 1) discloses additional steps in a specific example (depicted in FIGS. 10-16) of scanning the nodes and links of a graph in order to enable the graph to invoke computations directly. While execution of a graph is performed, there is no description that such a graph corresponds to the "modified graph" with runtime determined parameters as described in elements (a) – (f) of claim 1. Therefore, claims 1, 14, and 27 are patentable.

Independent claims 9, 22, and 35

The Examiner stated that the rejection of claim 1 applies to rejection of claims 9, 12, 14, 22, 25, 27, 35, and 38. However, none of the portions of Stanfill '072 cited by the Examiner with respect to the rejection of claim 1, or any other portion of Stanfill '072, teaches or suggests at least "(a) determining at runtime execution of the graph whether any component of the graph is defined as being a conditional component having an associated condition and an associated condition-interpretation; (b) evaluating the associated condition for every such conditional component; (c) modifying the graph at runtime execution of the graph in accordance with such evaluation and the corresponding associated condition-interpretation of at least one such conditional component by removing such conditional component and all connected flows to such conditional component from the graph before execution of the graph, based on an evaluation of the associated condition and the corresponding associated condition-interpretation for such conditional component." Therefore, claims 9, 22, and 35 are patentable.

Independent claims 12, 25, and 38

The Examiner stated that the rejection of claim 1 applies to rejection of claims 9, 12, 14, 22, 25, 27, 35, and 38. However, none of the portions of Stanfill '072 cited by the Examiner with respect to the rejection of claim 1, or any other portion of Stanfill '072, teaches or suggests at least "(a) determining at runtime execution of the graph whether any component of the graph is defined as being a conditional component having a an associated condition and an associated condition-interpretation; (b) evaluating the associated condition for every such conditional component; (c) modifying the graph at runtime execution of the graph in accordance with such evaluation and the corresponding associated condition-interpretation of at least one such conditional component by replacing such conditional component with a flow before execution of the graph based on an evaluation of the associated condition and the corresponding condition-interpretation for such conditional component." Therefore, claims 12, 25, and 38 are patentable.

Dependent claims 3-5, 8, 11, 13, 16-18, 21, 24, 26-27, 29-31, 34, 37, and 39

The dependent claims are allowable for at least the same reasons as those set forth above for the independent claims upon which they respectively depend.

Please apply any charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: 6-30-05

Elliott J. Mason, III
Reg. No. 56,569

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110-2804
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21064125.doc